1. Using the operator precedence table above, evaluate each expression and state what gets printed.

```c
int x;
int a = 11;
int b = 4;

x = a % b * 3 - b + a;
printf( "%d\n", x );
```

(3 pts)

```c
int x;
int a = 11;
int b = 4;

x = a / 2 + a + 2 * b;
printf( "%d\n", x );
```

(3 pts)

2. What gets printed in the following blocks of statements?

```c
int a = 5;
int b = 8;
int c = 15;

if ( (b > a) && (b != 8) || (c == b+10) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)

```c
int x = 3;
int y = 2;
int z = x - 9;

if ( (z > 6) || (x >= y) && (z > y) )
    printf( "True" );
else
    printf( "False" );
```

(3 pts)
3. Which of the following are not valid C identifiers? (Circle your answer(s).) (4 pts max)  
[0 if all circled; -? for each incorrect]

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLOAT</td>
<td></td>
</tr>
<tr>
<td>A_PhD_In_Swing</td>
<td></td>
</tr>
<tr>
<td>2_Degrees_In_BeBop</td>
<td></td>
</tr>
<tr>
<td>A_Master's_In_Rhythm</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td></td>
</tr>
<tr>
<td>A.F.I.</td>
<td></td>
</tr>
</tbody>
</table>

4. Fill in the blanks with the appropriate types and format specifiers to output the values correctly. (6 pts)

```c
void main( void )
{
    __________ a = 98.6;
    __________ b = '9';
    __________ c = 420;
    printf( "b = %____
             nc = %____\n             na = %____\n            ", b, c, a );
}
```

5. What gets printed? (9 pts)

```c
void main( void )
{
    int num = 3;
    switch ( num = num + 2 )
    {
        case 5:
            printf( "A
            
            num = num + 2;
            case 7:
            printf( "B\n            
            num = num + 4;
            break;
            case 3:
            printf( "C\n            
            num = num + num;
            break;
            case 3:
            printf( "D\n            
            num = num + 5;
            default:
            printf( "E\n            
            num = num + 8;
            break;
            printf( "num = %d\n", num );
    }
```
6. Write an equivalent for loop for the following while loop. (12 pts)

```c
i = 21;
while ( i >= j )
{
    j = i / j;
    printf( "%d %d\n", i, j );
    --i;
}
```

7. What gets printed in the following block of statements? (8 pts)

```c
#define SIZE 8
int i;
int array[SIZE] = { 11, 7, 3, 4, 6, 2, -1, 5 };
for ( i = 0; i < SIZE; ++i )
    if ( i % 2 == 1 )
        printf( "%d\n", array[i] );
```

8. (12 pts)

```c
#include <stdio.h>
#define SIZE 8
void function1( int var1, int var2 );

int main( void )
{
    int a[SIZE] = { 11, 7, 3, 7, 6, 2, -1, 3 };
    int z;
    scanf( "%d", &z ); /* Read an integer as input */
    if ( a[z] < a[z+1] )
    {
        z = 3;
        function1( a[z], z );
    } else {
        z = 7;
        function1( z, a[z] );
    }
    return 0;
}

void function1( int var1, int var2 )
{
    int i = 0;
    do {
        printf( "%d", var1 );
        ++var1;
        ++i;
    } while ( i < var2 );
}
```
9. What gets printed? (16 pts)

```c
#include <stdio.h>
#define SIZE 7

int janet( int x );

int
main( void )
{
    int array[SIZE];
    int i;
    for ( i = 0; i < SIZE; ++i )
    {
        array[i] = janet( i );
    }
    for ( i = 0; i < SIZE; ++i )
    {
        printf( "%d\n", array[i] );
    }
    printf( "i = %d\n", i );
    return 0;
}

int
janet( int x )
{
    int number[SIZE] = { 15, 23, 34, 42, 50, 67, 76 }; 
    if ( x < SIZE / 2 )
        return ( number[x] / 10 );
    else
        return ( number[x] % 10 );
}
```

Use the following to answer the questions below: (5 pts)

A) Actual Argument
B) Function Definition
C) Function Prototype (Function Declaration)
D) Return Type
E) Local Variable
F) Formal Parameter
G) C Preprocessor Directive

What is `array` in line 10?  

What is line 5?  

What is the `#include` in line 1?  

What is `x` in line 29?  

What is the `int` in line 28?  

What is the `i` in line 12?  

What is the `array` in line 10?  

What is the `SIZE` in line 4?  

What is the `main` in line 7?  

What is the `janet` in line 29?  

What is the `return` in line 25?  

What is the `printf` in line 18?  

What is the `for` in line 13?  

What is the `i` in line 14?  

What is the `i` in line 15?  

What is the `printf` in line 21?  

What is the `array` in line 14?  

What is the `SIZE` in line 11?  

What is the `printf` in line 23?  

What is the `printf` in line 24?  

What is the `printf` in line 22?  

What is the `printf` in line 19?  

What is the `printf` in line 17?  

What is the `printf` in line 16?  

What is the `printf` in line 26?  

What is the `printf` in line 20?  

What is the `printf` in line 25?  

What is the `i` in line 18?  

What is the `i` in line 19?  

What is the `i` in line 22?  

What is the `i` in line 23?  

What is the `i` in line 24?  

What is the `i` in line 27?  

What is the `i` in line 28?  

What is the `i` in line 29?  

What is the `i` in line 30?  

What is the `i` in line 31?  

What is the `i` in line 32?  

What is the `i` in line 33?  

What is the `i` in line 34?  

What is the `i` in line 35?  

What is the `i` in line 36?  

What is the `i` in line 37?   

```c
#define SIZE 7
```

```c
int janet( int x );
```

```c
int
main( void )
{
    int array[SIZE];
    int i;
    for ( i = 0; i < SIZE; ++i )
    {
        array[i] = janet( i );
    }
    for ( i = 0; i < SIZE; ++i )
    {
        printf( "%d\n", array[i] );
    }
    printf( "i = %d\n", i );
    return 0;
}
```

```c
int
janet( int x )
{
    int number[SIZE] = { 15, 23, 34, 42, 50, 67, 76 }; 
    if ( x < SIZE / 2 )
        return ( number[x] / 10 );
    else
        return ( number[x] % 10 );
}
```